

# **Path Programable Logic:**

## **A Structured Design Method for Digital and/or Mixed Analog Integrated Circuits**

B. Taylor  
Bonneville Microelectronics Inc.  
1399 S. 700 E. Suite 10  
Salt Lake City, Utah 84105

### **Abstract-**

The design of Integrated Circuits has evolved past the black art practiced by a few semiconductor companies to a world wide community of users. This was basically accomplished by the development of computer aided design tools which were made available to this community. As the tools matured into different components of the design task they were accepted into the community at large. However, the next step in this evolution is being ignored by the large tool vendors hindering the continuation of this process. With system level definition and simulation through the logic specification well understood, why is the physical generation so blatantly ignored. This portion of the development is still treated as an isolated task with information being passed from the designer to the layout function. Some form of result given back but it severely lacks full definition of what has transpired. The level of integration in I.C.'s for tomorrow, whether through new processes or applications will require higher speeds, increased transistor density, and non-digital performance which can only be achieved through attention to the physical implementation.

## **1 Introduction**

The future of circuit integration is rapidly moving toward high speed and/or analog elements. Digital signal processing techniques can take advantage of the fastest possible logic and with the Gallium Arsenide process becoming available this means clock rates of 250-500 Mhz. The "real world" is also inherently analog in nature and this means more system integration will require analog operations. Both of these areas require absolute control of the physical placement of transistors and interconnect.

Tool makers have historically treated physical layout almost as an after thought in the design process. The conventional design methods ignore the physical placement until the end and can then only define it in a geometrical format with no real relationship to the original definition except for connectivity. There have been tools developed to help manage this problem such as Frameworks but nothing to "fix" the problem.

The way to correct this problem area is to place equal importance on the physical as well as the logical and system definition. Equal importance means that they are all done at the same time in the design process and best by the same person. Adding this third dimension cannot be realized unless the design method is changed to allow it.

Physical definition fundamentally contains positional information and effects the performance and possibly the function of transistors. In this new design method the designer can take advantage of this positional information by defining what logic elements are to be adjacent and how interaction, whether connection or function, is to take place on all sides. To give equal importance to the interconnect and simplify its implementation would mean to have it on the same level as logic functions. Implementing logic is having predefined transistors personalized at a local position. Applying this equally means having predefined wire interconnects with personalized connectivity at a local position.

The requirement for this new design method clearly could not be satisfied by current tools. The benefits and demands for such a method could no longer be ignored. This led to the development of a new VLSI design tool called Path Programmable Logic (PPL) [1] [2].

## 2 The PPL Design Method

PPL is a two dimensional specific cellular grid-based design method which combines logic function and physical layout definition at a higher level. This higher level relates to a system level description in terms of binary values, register descriptions, logic operations, as well as connectivity. In the case of analog operations, it can utilize opamps, switches, comparators, and any other components that can be realized on an I.C. These functions are represented as a character. The character has a one to one correspondence of position, form factor and area to both its logic icon and physical implementation on the chip. Each cell with connecting wires running through all four sides is placed into a "sea-of-wires" through the use of these characters. The connection between adjacent cells is implicitly made unless a break is inserted to electrically isolate them from each other.

The PPL method places more emphasis on wiring than on transistors. The PPL circuit plane is covered with two sets of wires, one running horizontally, the other running vertically as shown in Figure 1. Cells are placed under wires to form logic function. Subsets of the wires in each direction are collected into an area called a unit cell. The size of the unit cell is determined by the number of wires that must pass through in each direction and the number of transistors for forming the simplest function.

PPL cells are designed to join all four sides with interconnecting wires running through the edges of each cell at the predefined wire locations. Interconnection between adjacent cells is implicitly made, unless a break is inserted. Breaking a wire which comes from a port of a cell means that port of the cell does not connect to the outside world as shown in Figure 1. Studies have shown that more connections are made than breaks of connections.

The PPL design method simultaneously specifies logic, layout, and interconnect since the spatial mapping of the PPL character to logic symbol or physical layout is one-to-one. Thus, like full custom design, the designer has control over placement and routing which effect signal coupling, inductance and stray capacitance, while, like semicustom design, hiding layout details from designer. This feature is ideal for the design of analog and high speed, critical path circuits.

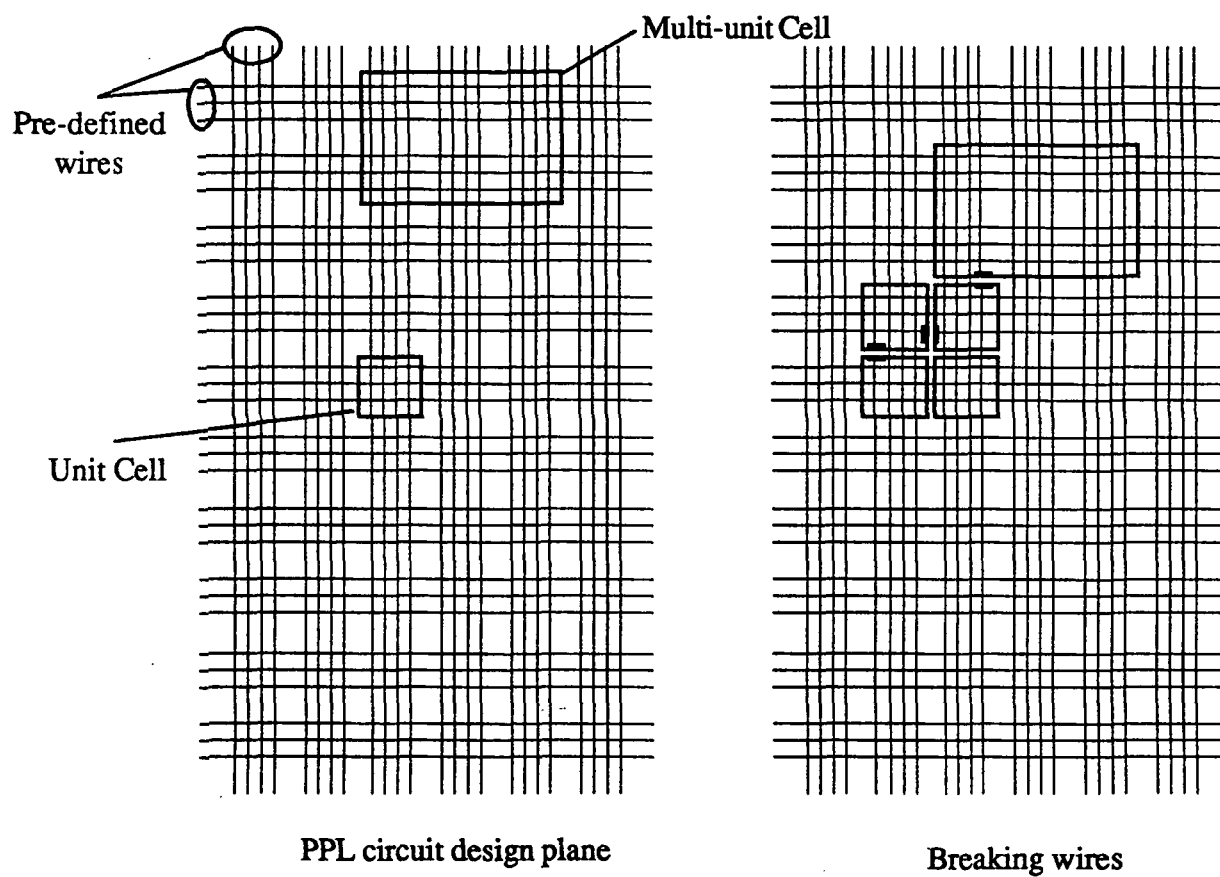


Figure 1: PPL circuit desing plane & Breaking wires

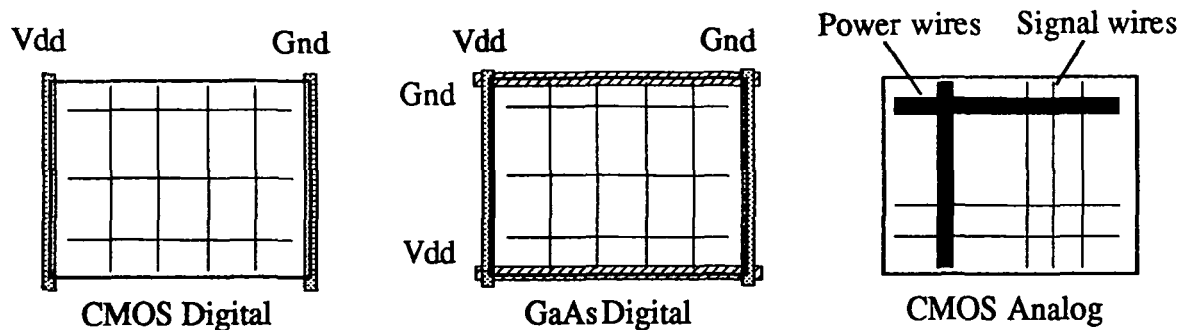


Figure 2: Varied implementation for power distribution

The power busses can be included in the unit cell (Figure 2) so that no explicit power and ground need to be done by the designer. The configuration can be changed according to the need such as alternating in the Y-axis for lower power Cmos logic, in both X and Y directions forming a mesh in different layers of metal for high speed, high current GaAs logic, or routeable for Analog implementation. Power busses can also be shared between cells in row or columns which implies that cells at alternate row or columns are being mirrored automatically by the design tool.

### 3 PPL Cells

With an initial design plane filled with blank cells, PPL circuits are designed by replacing blank cells with cells that form logic functions. In Figure 3, one blank cell is replaced by a cell connecting a column wire to a row wire. Another blank cell is replaced by an unit capacitance cell. All cells are rectangular. Ports at four edges of a cell correspond to the points where signal wires or maybe power wires (as in the case of analog circuit section) cross cell borders. Cells can be as small as a unit cell or a multiple unit cell size in both x and y directions. Cells may have modifiers that may change a cell wire scheme, its function, or its value. For instance, cell "c" (capacitor cell) with modifier of 0 means its capacitance equals 1.00 unit capacitance (Cu), while a modifier of 99 means its capacitance is 1.99Cu.

Digital cells are quite primitive, consisting of 2 to 4 transistors and covering only one grid location (unit cell size) in many cases. They have been carefully designed to fit together to form a useful distributed logic function. Figure 4 shows a PPL circuit of an EXOR function which looks very similar to the format in the EXOR truth table. Each PPL cell is represented by a logic character with specific meaning. For example, the logic symbol 1 indicates that this "1" cell senses a column containing a logic Q or TRUE output.

The 1 and 0 characters will actually group when they are on the same row forming an AND logic function. The plus characters will group in the column or vertical direction forming an OR function. The design tool will automatically complete the gate structure by connecting to power and, in the case of complementary CMOS logic, adding the appropriate connections to each output forming the node and the series the N-channel transistors are

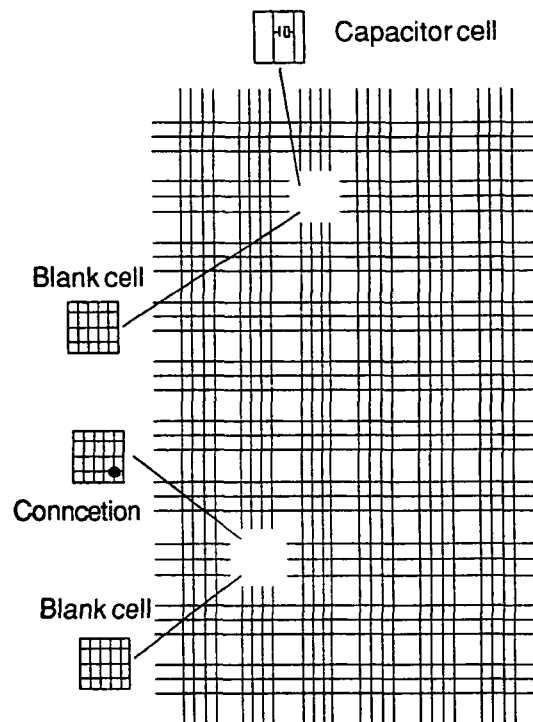


Figure 3: PPL Design-replacing blank cells

grounded at one end and shorted to the node on the other. This expression is then extracted as gates in a netlist format complete with performance characteristics annotated to the gates because the physical layout is also defined when the character is placed.

A	B	Out	EXOR truth table		
1	1		A	B	Out
1	0	+	1	0	1
0	1	+	0	1	1
PPL circuit					

Figure 4: PPL Circuit and truth table for EXOR

## 4 PPL Design

**Notation** The data entry in this new design method required a different type of notation, one that could be handled at a higher level than logic gates, void of too much information but easily understood as to its construction and function. With the interconnecting being

Characters	22
Transistors	190
Area	150 mi. sq. @ 2u rules
Description File	670 bytes
Schematic File	25173 bytes
Layout Cif file	20447 bytes
Gate Netlist	3737 bytes
Transistor Netlist	11764 bytes

Table 1: Counter Statistics

fixed, this notation did not need to explicitly display it. Removing interconnect from a drawing greatly reduces its complexity and removes a undesired burden from the designer. With a coarse grid defined by the unit cell placements an icon representation equal to this can be text characters. To show size of cells, a character with fill characters occupying the extent of a large cell can be displayed. The use of characters is also more translatable to a system level description.

```

      | 1   . 0   0   1  +|
      .. .. .. .. ..
      |" " " " " " " " +
      " " " " " " " " |I
      " " F " F " F " F " |
=4="r"r   r   r   r   .1 |
      .. .. .. .. ..
      U " U " U " U " a |

```

Figure 5: Design notation of counter

Figure 5 is the design notation for a BCD counter. It is constructed of 4 basic flip-flops ("F" characters occupying 4 rows by 2 columns) placed next to each other causing them to be connected together. The "4" character generates a two phase clock driver for the flip-flops. A synchronous reset is added to the flip-flops through the use of the "r" character. The flip-flops are made to count by adding the "U" character. A data path in the vertical direction is constructed in this manner and its width is a simple expansion or repetition in the horizontal direction. The binary value of 1001 (nine) is decoded at the top and ORed with an external reset signal then sensed by a "1" character that drives the synchronous reset of the flip-flops. This reset is also ANDed with an external count enable signal by the "a" character to control the carry-in of the "U" cells. This notation completely defines the function and placement of 190 transistor occupying 88 (8 × 11) unit cell locations. The size of the physical layout would now depend upon the technology to be used ie. Gate Array, GaAs, Custom, 2u rules, 1.2u rules, etc. This same notation is transportable across any of these technologies without any redefinition, only a reverification of performance.

The design notation requires only 22 character placements to fully define the BCD counter. This represents the amount of actual effort required by the designer. The con-

nectivity was generated automatically and for clarity some signal wire names were added that resulted in the description file being 670 bytes of data. This file contains enough information to allow the automatic generation of the schematic, layout, and annotated netlists. Table 1 of counter statistics shows designer's effort is leveraged approximately 3000X.

## 5 Conclusion

IC's which contain analog or high speed sections require the design tools to give the designer the ability to place the interconnections as well as the logic. The new PPL design method incorporates tools that meet this requirement. No other tools, aside from the old methods of handcrafting a custom chip, provide this capability. Other considerations include transistor density, productivity, and design portability. PPL designs are typically within 10% of custom densities, are completed in one tenth the time of custom or one half the time of standard cell implementations and are readily ported to various technologies and/or design rules. Total design effort can be easily done by a single designer without need of scheduling and interfacing with specialized resources such as layout engineers. The critical aspects of the design are always taken into consideration because the person defining the critical areas is also doing the implementation. The data compression is sufficient to allow designs to incorporate millions of transistors in a manageable form. The mundane but critically important task of documentation is taken care of by the concurrent generation of schematics from the logic placement.

## References

- [1] K.F. Smith, B.E. Nelson, T.M. Carter and A.B. Hayes, "Computer-aided design of integrated circuits using path-programmable logic," IEEE Electro '83 Professional Program Session Record, 1983.
- [2] B.E. Nelson, D.R. Morrell, C.J. Read and K.F. Smith, "The PPL integrated circuit design methodology," Computer Aided Design, 1986.